

**Поняття операції та виразу.
Пріоритет операцій. Арифметичні
операції. Операції над рядками. Поняття
операції та виразу. Пріоритет операцій**

Візуальне середовище дозволяє відносно просто виконати красивий, наділений великими можливостями графічний інтерфейс. Однак мета програми полягає в розрахунках, аналізі, порівнянні й організації зберігання вихідних даних. Для цього доведеться писати відповідні оператори, основною частиною яких стане обчислення виразів — арифметичних, логічних, рядкових. Цей розділ присвячений правилам побудови, обчислення й використання різних виразів.

Вираз — це комбінація символів, яка визначає порядок виконання дій над даними, що вводяться в комп'ютер, і складається з операндів, круглих дужок та знаків операцій. Операнди являють собою константи, змінні й виклики функцій. **Операція** — це дія, що виконується над операндами. Наприклад, у виразі $a * \sin(x)$ пропонується обчислити значення синуса функції від змінної x , а потім помножити результат на a .

Операції в мові Delphi поділяються на арифметичні, операції відношення (порівняння), логічні (булеві), операції над рядками, одержання адреси тощо. Вирази відповідно називаються арифметичними, виразами відношення (порівнянними), логічними (булевими), рядковими тощо залежно від того, якого типу операнди та операції в них використовуються.

У складних виразах порядок обчислення значень позначається дужками, а якщо поспіль ідуть кілька операцій, не розділених дужками, то порядок обчислення визначається пріоритетом (старшинством) і асоціативністю операцій. За пріоритетом операції поділяються на чотири рівні (табл.7.1).

Таблиця 7.1

Пріоритет операцій у мові Delphi (Object Pascal)

Операція	Пріоритет
not	Перший (найвищий)
*, /, div, mod, and, shl, shr, as	Другий
+, or, xor	Третій
=, <>, <, >, <=, >=, in, is	Четвертий (найнижчий)

Операції з вищим пріоритетом виконуються раніше, ніж операції з нижчим пріоритетом. Якщо ж поспіль ідуть кілька операцій із рівним пріоритетом, то порядок обчислення визначається асоціативністю, яка в усіх операцій в Object Pascal однакова — зліва направо. Нижче в табл. 7.2 наведено приклади виконання операцій відповідно до зазначених правил.

Таблиця 7.2

Приклади виконання операцій

Вираз	Порядок обчислення	Пояснення
$a - b + c$	$((a - b) + c)$	Усі операції одного рівня пріоритету
$a * b / c$	$((a * b) / c)$	Усі операції одного рівня пріоритету
$a * b + c / b$	$(a * b) + (c / b)$	Операції * і / мають вищий пріоритет, ніж операція +
$a = b \text{ or } c = d$	$(a = (b \text{ or } c)) = d$	Операція or має вищий пріоритет, ніж операція = . Після виконання or операції = виконуються зліва направо

Арифметичні операції

Найчастіше в програмах використовуються арифметичні операції; вони виконують арифметичні дії над значеннями операндів цілочислових і дійсних типів даних (табл. 7.3).

Таблиця 7.3

Позначення арифметичних операцій в Delphi

Операція	Дія	Тип операндів	Тип результату
+	Додавання	Цілий, дійсний	Цілий, дійсний
-	Віднімання	Цілий, дійсний	Цілий, дійсний
*	Множення	Цілий, дійсний	Цілий, дійсний

/	Ділення	Цілий, дійсний	Дійсний
div	Цілочислове ділення	Цілий	Цілий
mod	Остача від ділення	Цілий	Цілий

Операції додавання, віднімання та множення відповідають аналогічним операціям у математиці. На відміну від них, операція ділення має три види: звичайне ділення (**/**), цілочислове ділення (**div**), остача від ділення (**mod**). Результат, одержуваний унаслідок використання різних операцій ділення, проілюструємо прикладами:

Вираз	Результат	Вираз	Результат
-12 / 4	-3.0	12/5	2.4
25 div 7	3	6 div 8	0
25 mod 7	4	6 mod 8	6
—10 div 4	-2	20 mod 5	0

Примітка. Логічні (булеві) операції та операції відношення (порівняння) буде розглянуто в наступному розділі.

Операції над рядками

Рядковий тип даних — один із найчастіше використовуваних у візуальних програмах. Багато в чому це зумовлене тим, що більшість відео-компонентів для введення даних (навіть числових) працюють із рядками.

Рядок — це послідовність символів. У Delphi (Object Pascal) існує кілька типів рядків. Основні з них наведено в табл. 7.4.

Таблиця 7.4

Основні типи рядків Delphi

Тип рядка	Максимальна довжина	Використовувана пам'ять	Сфера застосування
ShortString	255 символів	Від 2 до 256 байт	Погана сумісність, зберігання невеликих рядків, швидка обробка
AnsiString	Близько 2 ³¹ символів	Від 4 байт до 2 Гбайт	8-бітні символи (ANSI), DBCS ANSI, MBCS ANSI та ін.

WideString	Близько 2 ³⁰ символів	Від 4 байт до 2 Гбайт	Юнікод-символи - багатокоористува- цькі сервери, мультимовні додатки
------------	-------------------------------------	--------------------------	--

Для більшості цілей підходить тип Ans і String.

Розглянемо основні стандартні функції та процедури оброблення рядків.

1. Функція ***Length(Str: String)*** повертає кількість символів (довжину рядка). Приклад використання функції:

```
var
  Str: ansistring; L: integer;
begin
  Str: = 'Happy New Year!';
  L:=Length(Str); {результат L = 15}
end;
```

2. Процедура ***SetLength(Str: String; NewLength: Integer)*** може зменшувати довжину рядка. Якщо рядок містив більшу кількість символів, ніж задано в другому параметрі процедури, то «зайві» символи обрізаються. (Нагадаємо, що функція повертає обчислене значення у своєму імені й може використовуватись у виразах, а процедура передає результат списком параметрів і використовується як самостійний параметр.) Приклад використання процедури:

```
var Str: ansistring;
begin
  Str: = 'Happy New Year!';
  SetLength(Str, 5); {результат Str = 'Happy'}
end;
```

3. Функція ***Pos(SubStr, Str: String)*** повертає позицію підрядка в рядку. Нумерація символів починається з одиниці. У разі відсутності підрядка в рядку функція повертає значення 0. Приклад використання функції:

```
var Str1, Str2: ansistring;
  P: integer;
begin
  Str1:='Happy New Year!';
```

```
Str2:='New';  
P:=Pos(Str2, Str1); {результат P = 7}  
end;
```

4. Функція **Copy(Str: String; Start, Length: Integer)** повертає підрядок (частину рядка) починаючи із символу **Start** завдовжки **Length**. Обмежень на параметр **Length** немає, однак якщо числове значення перевищує кількість символів починаючи з позиції **Start** до кінця рядка, то рядок копіюється до самого кінця. Приклад використання функції:

```
var Str1, Str2: ansistring;  
begin  
  Str1:='Happy New Year!';  
  Str2:=Copy(Str1, 7, 3); {результат Str2 = 'New'}  
end;
```

5. Процедура **Insert(SubStr: String; Str: String; Pos: Integer)** вставляє в рядок **Str** підрядок **SubStr** у позицію **Pos**; починаючи з позиції вставки літери вихідного рядка розсуваються. Приклад використання процедури:

```
var Str: ansistring;  
begin  
  Str:='Happy Year!';  
  Insert('New ',Str,7); {результат Str='Happy New Year!'}  
end;
```

6. Процедура **Delete(Str: String; Start, Length: Integer)** видаляє з рядка **Str** символи починаючи з позиції **Start** завдовжки **Length** (якщо вказаний параметр виходить за межі рядка, видаляє символи до кінця рядка). Приклад використання процедури:

```
var Str: ansistring;  
begin  
  Str: = 'Happy New Year!';  
  Delete(Str, 6, 250); {результат Str = 'Happy'}  
end;
```

7. Функції **AnsiUpperCase(Str: String)** і **AnsiLowerCase (Str: String)** переводять рядок відповідно у верхній і нижній регістри (широко відомі в літературі функції **Uppercase** і **LowerCase** виконують ті самі дії, але тільки для рядків із латинських букв). Приклад використання функції:

```
var Str1, Str2, Str3: String;  
begin Str1:='Задача';  
    Str2:=AnsiUpperCase(Str1); { Str2 = «ЗАДАЧА» }  
    Str3:=AnsiLowerCase(Str1); { Str3 = «задача» }  
end;
```

Рядки можна порівнювати між собою звичайними операторами порівняння:

```
var Str1, Str2, Str3: ansistring; logic1, logic2: boolean;  
begin  
    Str1:='Cat'; Str2:='Cat'; Str3:='Dog';  
    logic1:=(Str1 = Str3); {результат logic1 = False }  
    logic2:=(Str1 = Str2); {результат logic2 = True }  
end;
```

Якщо рядки ідентичні, логічний вираз стане *True*.

ТЕОРЕТИЧНІ ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Дайте визначення поняття «вираз».
2. Що собою являють операнди та операції?
3. Які типи операцій використовуються в Delphi?
4. У якому порядку за пріоритетом виконуються операції в мові Delphi?
5. Наведіть приклади арифметичних операцій у Delphi.
6. Назвіть основні типи рядків Delphi.
7. Перелічіть основні стандартні функції та процедури оброблення рядків.